Learning Chinese

Rod Smith
associative-ai.com

21 March 2019

Abstract

Recent progress in "narrow AI" has secured growing acceptance of the technology in industry and elsewhere. Yet AI's original goal, human-like general intelligence in a computer (AGI), seems as elusive as ever. Philosopher John Searle's Chinese Room Argument (CRA) including thought experiment purports to show that AGI is fundamentally untenable. However, AI tends to ignore this conclusion. I reason that Searle's argument is extremely important. It doesn't show that computers will never think. It shows that some AI concepts are flawed. For a start, they stop the room learning Chinese (or anything). Once the room's flawed computational ontology is corrected, it can learn and build inner semantic structures. If true, this both defeats the CRA and suggests a path ahead.

Summary of arguments

Searle's Chinese Room Argument (CRA) conclusion follows from its premises. But the truth of a key premise is challenged. This assumption comes in two forms: (1) computers are purely symbol-manipulating devices; (2) computers are purely syntactic devices. (1) is challenged by arguing that computers can internally manipulate (create, modify, move, delete...) items other than symbols. (2) is challenged by arguing that instances of this other type of item combined with symbols can have intrinsic semantic content.

Overview

The first half of the present text examines concepts of the research field of Artificial Intelligence (AI) including that of the electronic digital computer. How accurate are these ideas? And if inaccurate, does the inaccuracy matter? Searle's Chinese Room Argument is introduced.

This introduction includes discussing the idea of computation as applied to computers, and discussing relevant work of the idea's creator, Alan Turing. Further, having considered the computational view of the machine, an alternative associative view is outlined.

The second half of the present text uses alleged insights gleaned in the first half to focus a bright light on key aspects of the Chinese Room Argument including thought experiment. An attempt is made to discover what is fundamentally wrong with the CRA. The room is supposed to be a computer programmed with a computation of human-like general intelligence. Why does it fail to first try to learn Chinese before trying to understand the Chinese language? A consideration of this question suggests a rebuttal of the CRA. I present this claimed rebuttal then consider what implications it has, if sound and valid, for future AI research.

Introduction

According to AI theorist Stevan Harnad in 2001, legendary co-author of the 1969 paper[1] introducing the frame problem, Patrick Hayes, proposed that the field of cognitive science be redefined as "*the ongoing research program of showing Searle's Chinese Room Argument* (CRA) *to be false*"[2].

CRA specialist, David Cole, in 2004 reported:

> "*the Chinese Room argument is probably the most widely discussed philosophical argument in cognitive science to appear in the past 25 years*"[3].

Searle, in 2014:

> "*the project of creating human intelligence by designing a computer program that will pass the Turing Test, the project I baptized years ago as Strong Artificial Intelligence (Strong AI), is doomed from the start*"[4].

Summary in 2019, Chinese Room list at PhilPapers.org, edited by Eric Dietrich:

> "*The Chinese Room argument, by John Searle, is one of the most important thought experiments in 20th century philosophy of mind. The point of the argument is to refute the idea that computers (now or in the future) can literally think*".

Today, the CRA, though disbelieved by many in AI, has not been decisively refuted and remains a resilient attack on the field's original goal of human-like general intelligence (AGI) including sensory perception in a computer.

Whether or not we agree with the CRA's conclusion, the CRA is a preeminent argument based on concepts used today to understand computation, the mind, and the digital computer. It's clearly in everyone's interest, both advocates of AI and opponents of AI, to discover whether these concepts are up to the task of making the greatest discovery of all time – how to create a genuinely human-like mind including perception in a machine.

Core concepts of Western alchemy were fervently embraced for over 2,000 years since at least Empedocles' 5th Century BCE proposal of the four elemental substances designated earth, air, fire and water. But these concepts proved inadequate for understanding the nature of supra-atomic substance. Are today's concepts of AI and cognitive science adequate for understanding the nature of the digital machine mind?

This is the main utility of the CRA: it puts these concepts to the test. And the main conclusion of the CRA: they fail the test. The CRA doesn't show that computers will never think. It shows that we need better concepts for understanding the device for historical reasons called a computer.

Chinese Room thought experiment

In the thought experiment, a group of Chinese speakers gather outside a sealed and windowless room and write Chinese question on cards, one ideogram per card. They push these symbols in the correct order into the room through a slot in the door. A man in the room has a rule book in English that he follows to the letter without deviation. There are baskets of spare Chinese symbols on the floor. The rules say things like: If the input symbol has the shape <*example or description of a shape goes here*> then go find a symbol of shape <*example or description of a different shape goes here*> and then go do such-and-such with the found symbol. Sometimes the rules require the man to push symbols of certain shapes out the slot in the door.

The input symbols are sensible Chinese questions, and the output symbols, reasonable Chinese answers. But the man known no Chinese. In other words, he himself doesn't contain the meanings of the shapes. When processing the symbols, he doesn't access the meanings of their shapes. The rules refer only to the shapes. Their meanings are not present in the room. No meanings are present in the room. They never could be present because none are there at the start, and all the room gets is symbols, and the only reaction the room has to the substance of the symbol is to the substance itself or to the shape of the substance, and neither the substance nor its shape contains or caries or indicates the meaning of the shape.

Searle calls reacting to symbol shape without accessing the meaning of the shape, formal, or syntactic, reaction. A machine that can react only syntactically to the things it internally processes is a purely syntactic machine.

To an observer, the room is a fluent Chinese speaker and easily passes the Turing test for machine intelligence. But neither the man nor the room understand Chinese. The reason: symbol shape does not indicate the meaning of the shape. And all the room gets is the shape.

Analogously, the Room is (almost) a computer

Searle says the room presents the essence of the computer. The man is the CPU (central processing unit), the rule book is the program, and the Chinese symbols are the input and output.

Yet interestingly, any computer engineer knows that this picture is not entirely accurate. Computers don't react to the shapes of the things they internally process. Furthermore, humans can't perceive the things computers actually do internally process: clocked voltage levels, microscopic magnetic orientations, and semiconductor switch states.

Suppose a operating computer were dangled into the sea. It's highly unlikely even sharks could detect its clocked voltage levels. Sharks might detect electric fields, but the voltage levels in the data wires inside the machine typically change from one level to another millions of times each second. Perceiving these levels means reacting separately, according to the level, millions of times a second. Sharks certainly would seem to gain no particular advantage from evolving a capacity to detect by induction logic states in a computer data bus.

Perhaps this difference between the Chinese room and the computer is, as far as AI is concerned, irrelevant. Sure (so this rejoinder goes) humans can't see what computers process, and *can* see what the Chinese room processes, but that doesn't matter. Computers don't contain little men either, or rule books or baskets. Yet the Chinese room does, and the room is universally regarded as an accurate analogue of the computer. No objection to the CRA says that Searle got the analogy wrong. Both AI advocates and opponents agree that the analogy is relevantly accurate.

Well, until now. I argue that not only is it inaccurate, but the inaccuracy is hugely important. This is a statement of the inaccuracy: humans can perceive what the Chinese room processes but can't perceive what computers process. I hope to show that this inequality reveals a fundamental flaw in the CRA. This flaw relates to meanings.

Dire conclusion for AI

The CRA's main apparent conclusion can be summarized as follows. Whether computer input is created by humans (for instance by typing on a keyboard) or by sensors (in reaction to the proximate environment at the sensory surface), the problem is the same. A computer cannot acquire the meanings of the things it processes. It gets only the things themselves. And the things themselves do not contain, carry or indicate their meanings. However, having the meanings is necessary in order to understand the world including language. Hence, computers will never perceive or understand their environments.

They might easily pass the Turing test, but they will never think. They might behave intelligently, but they will never *be* intelligent. The quest for artificial intelligence in a computer is a fool's errand doomed to abject and disconsolate failure. *Strong AI is doomed from the start.*

Concept of Strong AI

The CRA is an attack on what Searle calls Strong AI which he explains as follows:

> "*...according to strong AI, the computer is not merely a tool in the study of the mind; rather, the appropriately programmed computer really is a mind.*", further, "*...many people ... are committed to the ideology of strong AI...*"[5]

Searle says this in his 1980 paper, *Minds, Brains, and Programs*, available online, in which he introduces the Chinese room argument. Still today, many AI researchers believe computers will one day think and have a mind in the same sense humans do. They believe in Strong AI to the extent this means computers will one day really think. In fact this belief inspires today's AGI research.

Weirdness

Searle uses the Chinese Room to argue against what he calls Strong AI. But let's take the counterfactual (if ... then ...). Assume Strong AI is true. If what Searle calls Strong AI is true, then the room, the computer, *will* understand the Chinese symbols. However, the room has never learned Chinese. Input to the room could be any language, so the room will understand any language, without having learned it, if what Searle calls Strong AI is true.

Even more freakishly, if the input symbols come from sensors, the room will understand its environment without having learned anything about it. What can we say about this? Well, this unlearned knowledge is extremely odd. Also, it's extremely un-human-like.

The Chinese Room is supposed to be an analogue of a computer programmed with what AI believes to be a human-like mind. But the room's unlearned knowledge shows that it is fundamentally not an analogue of the human mind. It's fundamentally *not* human-like.

Presumably Searle, in response, would happily say something like, The room can't intrinsically learn because this entails acquiring a semantics, and, for a fundamental reason that every CRA follower knows about, it can't do this. Furthermore, the purpose of the thought experiment is to make this reason clear.

However, we shouldn't blindly accept this rejoinder. The matter of learning Chinese still deserves to be examined, in my view. It's still important to ask: In Searle's description of the room, why doesn't the room first try to learn Chinese? That's what humans need to do. The second half of the present text tries to answer this question.

AI Concepts

The view that computers are for ever prisoners in a world of syntax is based on certain concepts of computation and of the computer (such as that the machines are purely syntactic devices). Are these concepts adequate? Perhaps, for a given purpose they are. But what about other purposes?

It's said that computers can only do what humans program them to do. For human use of the device, this idea makes admirable sense. Humans want a certain behavior, and the obvious way to get that is to program it into the machine.

Similarly, it seems sensible to use the concept of computing to understand a machine that people use for performing computations.

Yet human-like intelligence means developing behaviors not programmed in by an observer. It means acquiring behaviors through interacting with the environment. It means being marooned alone on an unfamiliar island, yet adapting and surviving. It means crash landing on a verdant alien planet in a galaxy far, far away, and surviving. So for a device that intrinsically learns and acquires human-like knowledge, is the concept of computing perfectly suitable? Or worse, is it even counter-productive?

A reply to this might be: the machine at issue was designed to perform computations, artillery tables, etc. Its original abstract theory of design is about computable numbers. Of course computation is the right concept with which to understand the machine. The use of the device is irrelevant.

Yet, because a device has a certain name or a certain design goal or a certain theory of design or a certain use, that doesn't necessarily mean it can't do other things. The simple operations of the machine that allow it to compute might not prevent it performing other types of processing, too.

Importance of the CRA

To me, the Chinese Room Argument is hugely important to AI and indicates salient problems with AI's theoretical foundations. Searle's attack is on AI. In order to effectively mount this attack he has to use AI's concepts. And he does this. But these concepts have problems. Searle doesn't see this and merely accepts the concepts. On the AI side, AI hasn't been able to conclusively rebut the CRA because it doesn't question its own concepts.

Notions about a machine's abilities are merely descriptions, merely ways to understand what the machine does and how it does it. The machine is the machine. It does what it does. Human understanding of the machine is another matter. The correct role of the CRA, its real importance, is to show that AI's understanding of the computer is less than perfect. An understanding that is negatively influenced by the machine's first and typical use and by its design goal. Computers will one day think. But in order to understand why and how, first we need to understand them better.

The computer as a purely syntactic device

The CRA says the meanings of the things computers process are forever out of reach of the machine (because it gets just the tokenized shape, and the tokenized shape says nothing about its meaning).

These meanings are assigned by humans. Searle:

> "*symbols, by definition, have no meaning (or interpretation, or semantics) ... except insofar as someone outside the system gives it to them*"[6].

Further,

> "*digital computers insofar as they are computers have, by definition, a syntax alone*"[7].

Here, by "*symbol*" Searle includes symbol shape. Meaning is assigned to the shape – by humans seeing the shape and giving it a meaning. More generally, a meaning is assigned to a value of a property. The property is shape, and the values are the different shapes.

So symbols, to Searle, have externally assigned meanings, and computers, to Searle, process symbols. But there's a problem here. Do the things computers internally process really have assigned meanings?

Do the things computers process have meanings?

As noted, humans lack the sensory apparatus needed to detect the things computers process. A human can perceive a certain shape and assign a meaning to it. But humans can't perceive clocked voltage levels. They might see a logic trace on a screen. That's merely a series of illuminated pixels in a screen. Humans can't perceive what computers process, and hence can't assign meanings to values of their properties. For biological reasons, the interpretive history of the symbol doesn't exist for what computers process.

It's often said that computers process 0s and 1s (which are linguistic symbols), and further, that sequences of these 0s and 1s or their ASCII forms (for instance words) have had meanings assigned to them. I thoroughly agree. It's very true that sequences of 0s and 1s and their ASCII forms have assigned meanings.

But computers don't process 0s and 1s. There are no items shaped "0" or "1" flowing through the wires of a computer. "0" and "1" are names chosen by tribal animals (and somewhat abstract names) of the things computers do in fact process, which are clocked voltage levels, magnetic orientations and semiconductor switch states. Certainly, the shapes of the names have assigned meanings, but it does not follow that the shapes of the things named also have assigned meanings. Or that values of any property of the things named have assigned meanings.

It's quite false to say that the things computers process are in the same referential boat as symbols. Contrary to what Searle says, the values of the properties of the things computers process actually have no assigned meanings at all.

Neither computers nor brains understand what they process

In other words, given what computers in fact manipulate, computers could never understand the meanings of these things, not because the meanings are out of reach, but because they don't exist in the first place.

Granted, a keyboard has shapes inscribed on its keys, screens can display shapes, and printers can print shapes, and these shapes can have meanings assigned by humans. But these shapes are merely encrustations on exposed surfaces of attachments of the machine put there by humans to facilitate human use of the device. To the machine itself, inside the device itself, the things processed there are meaningless. They have neither extrinsic nor intrinsic meaning. They have no assigned meaning at all.

But perhaps the computer itself could assign meanings to values of a property of the things it internally processes. But how would this happen? If it had human-like perception, it would have human-like sensors, and human-like sensors can't detect what computers process. Yet why not add an extra sense that detects the relevant clocked voltage levels? OK, but why bother? Humans can't perceive the neural electrical pulses in their own brains, yet are intelligent. Why should a computer detecting electrical pulse levels in its computer brain be relevant to computer intelligence?

Searle says a computer is a machine that processes objects that have values of a property, and that humans have assigned meanings to those values. Then he argues that because the computer gets only the instantiated values and not the assigned meanings, it could never think. But in reality, the values of the things computers process have no assigned meanings – just like the values of the things organic brains process. They have no assigned meanings, either.

Hence the situation is both better and worse than Searle maintains. It's worse because it's not that the machine is forever a prisoner in a universe of syntax, the semantics being eternally inaccessible. In fact no assigned semantics exist in the first place. It's better because neither does an assigned semantics exist for what organic brains process. If lack of a such a semantics is no impediment to human intelligence, well, perhaps it's no impediment to computer intelligence, either.

The idea that an intelligent system must understand assigned meanings of values of a property of the things it processes – and that intelligence derives from this understanding – to me seems totally specious. There's no homunculus examining the values of properties of the things processed. There has to be another answer. There has to be a better concept.

Searle's faulty concept of the computer

Searle doesn't explain that the things computers process have no meanings in the sense that linguistic symbols have meanings. To him, computers processes linguistic symbols. This is wrong. Humans use the symbol-processing idea to make it easier to think and write about the traditional human uses of the machined named after the first human use – computing. Searle's linguistic-computational idea of the machine conceals the fact that the things the device actually processes, unlike linguistic-computational symbols[8], have neither an intrinsic nor an extrinsic semantics.

Concepts predicated on human use are irrelevant

It's possible to abandon a concept about human use. For the computer, it's possible to abandon the explanatory concept of computation. In Turing's 1936 example[9], computation is something humans do with pen and paper by creating marks that they perceive and know the meaning of. Since the intrinsic nature of the computer is the key to genuine AI, concepts about an observer's use of the device seem quite irrelevant.

Does the classical concept of the computer allow intrinsic learning?

With the sort of computer program Turing talks about in his 1950 paper[10], a human dictates how the machine will react to different symbol shapes or groups of shapes, according to the human's understanding of the meanings of the shapes (motto: the intelligence is in the human, not in the machine). In other words, the human defines how the machine will respond to its inputs. But how might such a computer running a classical computational program intrinsically learn, if its behavior, or more generally its causality, is defined by an external cognate agent?

Turing, almost single-handed, developed then introduced the concepts now used to understand the computer. In his 1950 paper, he acknowledges the problem of intrinsic learning, and seeks to address this in various ways. These include with his extremely mysterious ephemerally valid rules[11], and in 1951, "indexes of experiences", which turn out to be descriptions typed into the machine by humans[12]. More abstractly, he suggests creating a simple child computer mind which is then "*subject to an appropriate course of education [to] obtain the adult [computer] brain*"[13], but he fails to give an insight into how this might be achieved. He predicts a form of connectionist "learning".

The computational and other AI-related concepts that Turing made famous and still underpin AI's understanding of the device were a *tour de force* of intellectual achievement, but that doesn't mean they are adequate for all possible purposes of the machine.

The two areas in which Turing seems to have the greatest difficulty are intrinsic learning and sensory perception. For example, even though sensory perception is crucial for human learning and intelligence, he doesn't mention it in his most widely read AI paper, "Computing Machinery and Intelligence", regarded as the mission statement of AI. And he mentions sense organs only once: "*It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English*". But there is no explanation.

If a machine is going to acquire a new behavior without being re-programmed, if it's going to acquire the behavior by itself, intrinsically, then there will need to be way for new rules, new causality, to be introduced *naturally* – meaning by nature rather than in accordance with the knowledge of an observing intellect. That is, introduced without the intercession of an external consciousness. But if the causation of the machine is a product only of human design (both hardware and software) then how is his possible?

If the classic concept of the machine is accurate, computers will never intrinsically learn. And since intrinsic learning is key to human intelligence, computers will never fully achieve human-like intelligence.

But is the classic computational idea of the computer accurate? People who prefer the (seeming) certainty of science might see no problem with today's classical understanding

of the computer: The facts are the facts. Concepts aren't even involved. Facts are involved. Computer Science knows how computers work. End of story.

Without wishing to be unkind, we have to realize that these people are modern alchemists. Alchemy actually gets a bad wrap. Ignoring the mystical aspects; ignoring the sun, Sol, the father, gold; ignoring the moon, Luna, silver, the mother; ignoring the doctrine of perfection of substance by the balance of the four elements, ignoring the quintessence embodied in the semi-catalytic philosopher's stone, ignoring all this, putting these and other mystical almost addictive ideas aside, if science is empirical investigation then the metallurgical theme of alchemy was almost pure science.

The amount of empirical experimentation and (encoded) record keeping in alchemical research was quite humongous. But as we now know, the concepts weren't good enough. Yet those poor concepts plus evolving equipment and experimental procedure laid the basis for chemistry and its extensive contribution to modern life. All that new chemical knowledge could have been used for great evil, and in certain cases it was. One use of Zyklon B by a government is well known. A use of wind-borne heavy gas by several governments is well known. But the overall result has been very positive.

Concept of associative processing

One concept with which to understand processing or manipulating substance inside a machine is *association*. In fact, the Chinese Room can process its inputs associatively. It can react just to association (a relation) between symbols that fall from the slot, rather than to their shapes (the values of a property).

One way the room can process associatively is simply to store incoming symbols in the order they drop from the slot. The temporal association of their entry one after the other into the room is permanently recorded as spatial juxtaposition in storage. In a sense, the storage removes the element of time, replacing it with a spatial relationship.

Another way of operating associatively is by reacting to the association in time of incoming symbols and storing alternate ones in different places on the basis of their arrival order, again without reacting to their shapes.

One could imagine a blind man hearing the thud of a symbol as it hits the floor of the Chinese room, then without identifying its shape, moving the symbol to the left, then for the next symbol to drop from the slot, moving it to the right, and repeating this alternation. In this way, he reacts to the temporal relation of association between the input symbols as they drop from the slot, but couldn't give a toss about their shapes.

These are simple examples. Perhaps they're trivial. But are there clearly non-trivial associative processes a computer could perform? And could these even be related to intelligence?

Computation by another name?

Yet is associating a form of computing or a part of the process of computing and hence not a separate and distinct concept?

Firstly, the process of associating seems to be a primitive. Computing, on the other hand, clearly entails several sub-processes. I think it would be generally agreed that associating

is atomic whereas computing is compound. In other words, associating is not computing by anther name.

But could it be a component of computing? And if so, if it's not part of any other process a computer could perform, then we would probably happily agree that associating is not a separate concept in its own right but merely part of the concept of computing.

However, if a computer could associate and perform a useful task without executing all the necessary steps of computing, then it would not be computing. It would be a concept in its own right with which to understand at least some abilities of the machine. A machine for historical reasons called a "computer", but a machine that could also usefully operate non-computationally.

Can a computer manipulate symbols without reacting to their shapes?

In the example of the Chinese room, the room can manipulate symbols without reacting to their shapes. We know that computers don't internally react to the shapes of anything, but rather react to values of different properties of what they internally process. Yet we adopt the fiction of reaction to shape to make it easier to talk and think (sometimes wrongly) about computers. So now adopting this fiction, can computers process symbols without reacting to their shapes?

At the level of human sequencing of simple computer operations, that is, at the level of typical programming, a computer can easily manipulate symbols without identifying their shapes. For instance, by executing the program line, B$ = INPUT$;. In this case, assuming a data length of one byte, the contents of the current input byte is stored in the memory location named B. (The dollar sign indicates an alphanumeric data type; the semicolon terminates the line.) The shape of the symbol received in input then stored at location B is unknown and irrelevant to the program line. The line simply stores current input, whatever its shape might be. *It treats all shapes the same way.* It doesn't distinguish one shape from another. Hence it's a very short and simple program line.

If the computer distinguished shape, there would need to be at least 256 instructions, one for each possible unique ASCII byte shape. If the two variables named "A$" and "INPUT$" named memory locations that could each hold a string of up to say 512 KB in length, more instructions would be needed than the estimated number of particles in the known universe.

But at the hardware level what happens? The process at the hardware level could be explained something like this. The input is generated, say, by a human pressing the "A" key on a keyboard. The keyboard circuitry transmits a code in clocked voltage form to the computer. This code is a sequence of clocked voltage levels. The computer's keyboard controller receives then converts this code into another sequence of clocked voltage levels that are often called the ASCII letter "A". In order to do this, the controller has to react to the "shape" (i.e., levels of the clocked voltages) in the input from the keyboard in order to create the right ASCII sequence of clocked voltages, which actually has the byte value name "01000001", or in clocked voltage levels, assuming the low value is called "0": low, high, low, low, low, low, low, high. There might be added clocked levels known as parity bits and stop bits.

So the keyboard controller has reacted to what, for convenience, we are calling shape. The ASCII clocked voltages are then loaded into a CPU register. To do this, the machine must

convert them into semiconductor switch states. In order to create the correct set of switch states, the computer must react to the "shape", that is the levels, of the ASCII voltages.

From the register, the switch states are converted back into clocked voltages and asserted on a data bus and then travel to memory where the memory controller converts them back into semiconductor switch states. So at this base level of computer functioning there's much reaction to shape (that is, to values of the relevant property of the substrate – voltage level, magnetic orientation, switch state). Even though at the programmer level, the level where human knowledge dictates the functioning of the machine, symbols can be manipulated completely blindly.

However, the base-level conversions are accidental in the sense they're needed because of the electronic nature of the machine. Other forms of the machine don't require such or any conversions. The Chinese room, for example, doesn't perform conversions. Symbols enter as input, then the very same instances of substance, the very same particular collections of atoms and molecules, are manipulated, stored and optionally expelled out the slot. (My argument of inaccuracy of the analogy doesn't concern conversion.) Conversion can be discounted, then, when considering the essence of the machine.

Natural association

According to the associative concept of interest, a human doesn't decide which shapes are associated together inside the computer. A human mind doesn't dictate the behavior of the machine according to what the human understands the shapes of the input and output to mean.

Some reaction to shape is actually acceptable to this idea of association. What's not acceptable is that a cognate creature, a designer, mandates the reactions of the machine to its inputs. The behavior of the machine needs to be mainly non-teleological – not designed.

Deep learning

The current AI method of deep learning is held out to have great promise, and is also gaining acceptance in the commercial world. A deep learning network is an artificial neural network (ANN) with multiple hidden layers.

Deep learning modules are used in self-driving vehicle AI systems, receive input from sensors, then contribute, often decisively, to autonomous vehicle action. But unfortunately, humans still determine machine behavior in deep learning vehicle systems.

An ANN trained on images is typically presented with thousands, hundreds of thousands or millions of images that depict one or more types of object, such as pedestrian crossing, bus, car, fire hydrant, pram, and pedestrian. Such images are usually first labeled by a human, that is, the relevant parts of the full image are circumscribed with graphical boxes which are then labeled "car", "bus", "bicycle", "pedestrian", and so on. Or a full image is labeled with the name of what appears to be the prominent type of object depicted in it.

Once a network is trained up, a human who knows what it has been trained up on, programs the AI system to respond in appropriate ways when the network signals a sufficient probability of "seeing" one of the trained-up-on macroscopic objects such as a pram or a pedestrian. This involves humans dictating the causation of the system on the basis of what the human understands the ANN to have identified.

Initial marketing hype about AI systems controlling self-driving cars has finally settled (thanks largely to corpses and legal cases). It's realized now that today's AI systems actually lack the perceptive abilities even of a door mouse. Deep learning removes the need to program minute detail (at non-trivial risk of serious error in identifying types of external macroscopic objects), but human knowledge still dictates system behavior for AI systems that control self-driving vehicles.

Deep learning system have potential but they also have high error rates in unrestricted driving environments, high power consumption, and "learn" by a very time-consuming process that is clearly not human-like. The lack of anything in humans approaching back propagation indicates that ANN's may not be the answer.

ANNs were founded on a 1940s' idea of association that is moderately complex and quite different from the primitive principle outlined in the present text. The 1940s' idea proposes an idealization of the biological neural process[14]. However, the principle outlined in the present text is completely divorced from any realization, and also is atomically simple.

Teleological v non-teleological theories

The theory of evolution is a far simpler explanation of life on Earth compared to the theory of divine design where every aspect of every creature is decided by cognate intention. The theory of human design of an AI system is very complex because the programmer designs (programs) every response in detail to every situation the machine might face. This computational approach to machine intelligence has well-known difficulties including the frame problem (McCarthy and Hayes, 1969[15]), problem of common-sense knowledge (McCarthy, 1959[16]; Minsky, 1968[17]), problem of combinatorial explosion (Lighthill, 1973[18]), symbol grounding problem (Harnad, 1990[19]) and Chinese Room Argument (Searle, 1980[20]).

Investigating alternative approaches is sensible. Turing's prediction, so regarded, of the computational machine with human-like general intelligence by the year 2000[21] was found to be wanting. No such device is on the horizon even today, two decades later.

The computer contains a very large quantity of uniquely addressable memory locations, and also a very high processing speed, relatively speaking. The computer is AI's development platform, and no viable alternative machine is on the horizon. If computers only compute, then why investigate alternative types of processing? But do computers only compute? No. In fact they can also associate.

Acceptable operations

Matching, for example, is a reaction to shape that is acceptable to the associative view of the device which for reasons of historical human use and design is called a "computer".

A matching instruction in the Chinese Room might be, Take the current symbol and go find a match among the piles of spares on the floor. This is perfectly acceptable. The same short instruction applies no matter what the symbol shape.

The matching process need not react to a whole symbol. If two equivalently located parts of a shape are found to be different, the matching fails. For a bit string, if the two bits in the first position are found to be different, the match fails and the operation returns a fail code. Match instructions are not different depending on the shapes of the symbols to be matched. They treat all shapes the same way. Hence the instruction is very short.

But if the instruction said, If the symbol is shaped <*example of a shape goes here*> then do <*such and such*>, but if the symbol is shaped <*example of a different shape goes here*> then go do <*something different*>, if the instruction said this, then it would not be acceptable to the present associative view.

Counting symbols of a given shape could be acceptable. Respective instruction would need to be generally couched. Such an acceptable instruction might say, Go to the list of distinct symbol shapes and find a match to the current symbol, and when a match is found, increase the respective count by 1, but if a match isn't found, add the current symbol shape to the list. With this instruction, the same simple rule applies, no matter the shape.

The list of different symbol shapes itself could be created, as indicated, without specifying shape by requiring that a shape be added if it is not already present (as determined by matching).

Where does difference reside? If the operations apply to all shapes equally without prejudice or favor, then the actual shapes that arrive in the input is where difference lies. The difference doesn't lie with the instructions. It lies with the shapes. With storage, a repository of the shapes received from sensors would exist. In this case, no human has decided which shapes are stored. The interaction of sensors with nature determines which shapes are dispatched to the central system, and how many. The human design of the sensor defines what are the possible shapes, but which of these and how many depends on the environment.

Repetition is quite OK. If a program says: If a shape is received over <*a threshold number of*> times in a processing period then store it, this is quite acceptable. The program line applies equally to all shapes. The difference is in which shapes actually arrive.

Associative view v. computational view

Human design makes the machine an automaton. And automatons can't intrinsically learn. A human is needed to change the causation of the machine. Turing in his 1950 paper:

> "*How can the rules of operation of the machine change? They should describe completely how the machine will react whatever its history might be, whatever changes it might undergo. The rules are thus quite time-invariant. This is quite true*"[22].

The associative view rejects the god-like notion of full human design. The computer needs *some* human-created rules. But not ones that define the shape of the output given the shape of the input. A computation defines the shape of the output by executing human-created rules about the shape of the input. In contrast, the associative idea says that the machine accumulates inner structure containing patterns of difference that reflect the sensed environment. Such structure will acquire and embody rules from its interaction with the sensed environment including the machine sensing the results of its own effector actions. Though to talk of "rules" as being embodied in such structure is probably a misleading way of thinking.

The human-created rules of the suggested associative approach are about building structure – not about what structure contains. Hence the human-created rules don't treat different shapes in different ways. To the human-created algorithms, input symbols are merely substance that populates structure. The substance *has* different shapes, so the structure is filled with difference, but the human-created rules ignore this difference. For

instance, there are no rules of the form: If the input = "A" then the Output = "B", or, If the input = "A" then do such and such.

Viewed this way, a human program facilitates knowledge and action but doesn't define it. The human can instruct that input be matched, or counted, or have a threshold count applied to it, or can instruct that the machine do various other equal-opportunity things. But not on the basis of input symbol shape.

The issue is the causation between input and output as embodied in inner structure. It's probably quite inappropriate to call this causation "rules". It's confusing at best. Perhaps the idea of rules should be limited just to printed or displayed listings of programs.

The associative view is quite different from Turing's published 1950 conception whereby the causation of the machine is fully human-designed. It's also quite different from the Turing machine description where most rules start with reaction to shape (scanning); and once the machine is set in motion, no input is allowed.

One thinks of a computer executing rules as that of the computer executing a program comprising symbols that a programmer has typed on a keyboard, seen on the screen, or printed on program listings. This is an very unfortunate view. The shapes in program listings are merely there to make it easy for humans to dictate the causation of the machine. The machine doesn't process the symbols of a program listing, or any symbols. To suppose they do is merely a convenient fiction that makes it easier to think about human use of the machine.

The idea that executing a program means processing symbolic rules is a flawed computational perspective on what happens inside the machine. Yet it seems useful to say that human-created programs, as printed out, are rules. They are rules to the programmer. It's just that executing programs are not, and never could be, rules to the machine. Just as neural processes are not, and never could be, rules to the human. There's no homunculus inside the organic brain.

Characteristics of association

The basic associative idea (as regards perception) simply says that sensor symbols that arrive together can then be linked in storage – associated. And also that there are a few simple equal-opportunity conditions to such linking and storing, which generate structure.

One happy effect of the approach of associating input is to make algorithms very small and simple, and consequently the processing of input very fast. Tests that I conducted in 1998 in both assembler and C showed a node-to-node processing speed, including processing the node, of less than 40 clocks, making a *cateris paribus* speed on a high-end desktop machine today, 20 years later, of probably over 100 million structural connector traversals each second. (Half jokingly, I called a single traversal a "signal propagation", or "sip", and measured (virtual) travel within the associative structure in sips per second, or just sips.)

We can say that the concept of associating, matching, counting matches, and applying a threshold match count is a concept about the intrinsic nature of the machine, since the behavior of the machine is not dictated by any external brain.

Most symbols (or rather what we are calling symbols which are not actually symbols) retained in the machine come from sensors. The behavior of a machine programmed in accordance with certain principles of association comes in part from sensory input in the

context of effector action, and the machine is definitely not a sophisticated simulacrum of the player piano or Le Canard Digérateur whose every action is a testament to the genius of human design.

Examining the Chinese Room

Returning to the Chinese room, this is a simple question. Humans have to first learn Chinese in order to understand it. In Searle's explanation of the room, why doesn't it first try to learn Chinese?

The answer is: It can't. Learning entails adaptation. The room can't adapt. Specifically, learning entails structural change (addition, modification, perhaps even deletion of structure). The Chinese Room has no structural ontological type. It has no elements with which to build inner structure. It has data to populate structure (it has symbols) but it has nothing to *relate* symbols together.

Unlike computers. Computers often build inner memory structures and populate them with data. They often relate units of data together, for example, with pointers.

A pointer is an address stored at a location. The address points to another location and any data that might be stored there (which might be another address)[23]. Self-reference is usually avoided because it implies a deadly embrace.

Searle's concept of the computer

Searle's concept of computation says computation is purely syntactic – that is, symbol manipulation according to rules about symbol shape but not meaning.

Searle also says that the devices called computers necessarily only compute. Hence the device must also be purely syntactic. Searle uses Turing's theory of the Turing machine to explain this. Computers were first designed to perform computations such as artillery tables.

But as noted, just because a device is designed to do X, and just because it is then used by humans to do X, and just because humans call it an Xer, that doesn't mean it can't do anything else. The idea of rules referencing symbol shape but not meaning is merely a human's idea of the human use of the machine. The idea of human use needs to be thoroughly abandoned.

Are computers purely symbol-manipulating devices?

The CRA says computers are purely symbol-manipulating devices. Searle (2014):

> "*...a digital computer is a syntactic machine. It manipulates symbols and does nothing else ... [The] programmed computer has syntax but no semantics*"[24].

In saying the "*programmed computer has syntax but no semantics*", he means that the machines react to the (what he calls) shape of what they process (which he calls symbols) and that this reaction does not and cannot include accessing the meanings (the semantics) of the shapes.

Defective ontology

More abstractly, Searle fixates on the room's reaction to values of a property. In the Chinese Room, the property is shape and the values are the different shapes. There is substance – the symbols. There is a property – symbol shape. There are values of that property – the different shapes. But there is nothing to *relate* instances of the substance together.

The room's ontology is defective. It needs more than just substance with properties. The ontology needs relationships, too. It needs entities that can connect symbols, or what amounts to the same thing – places where symbols are stored. As well as baskets of spare symbols, the room needs baskets of relational connectors. The room needs the components to build structure.

Are connective elements symbols by another name?

But are such relational elements of structure merely closet symbols? Does the concept of the symbol cover these connective elements? Or could the concept of *symbol* be reasonably and usefully expanded to include connective elements?

Connections have ends. The idea of a connection is actually the idea of a 2-term relation. The essence of a connection is its two ends. It's that by virtue of which an instance of the relationship exists and relates its terms. The terms are the two objects connected. In the Chinese Room, these objects are symbols. If the room had structural elements, these could connect two symbols together, thereby relating them.

However, what's between the two ends is quite immaterial to the connective relation *per se*. All that really matters, *per se*, is that if you're at one end, you can follow the substance of the relational element and get to the other end. You can get from one of the two connected particular objects, one of the two terms, to the other. In the Chinese Room, the connective elements could be lengths of string.

In fact in the typical computer realization of the 2-term connective relation, pointers, there is no direct physical link between the two ends. An algorithm at one end of the virtual connection typically uses the method of direct memory addressing to get to the memory location at the other end, and to the data, if any, stored there. This can happen remarkably quickly – perhaps in one or two clock cycles. Today's supercomputers could conceivably follow a billion such relational connections each second.

The utility of a symbol is its shape. Further, the essence of the symbol's utility is in difference (of shape). Whereas the essence and utility of a connector is unity – bringing objects together – connecting them, making a unity out of difference. Not destroying difference but joining difference together. In a sense the connector makes its terms one. It removes difference, or at least combines it into a singular item. It says there is something not different about the objects when related together, but there is something the same about them in the sense of belonging together.

In the Chinese room, connectors might be string, and each piece of string might have exactly the same values of properties as far as the room is concerned. Each piece might be identical, to the room. There might be no qualitative difference between each piece of string in the Chinese room, if it had connectors. In other words, connectors have a fundamentally different purpose from symbols. Symbols embody difference. Connectors unify rather than distinguish.

Hence, in terms of essence and utility, the relational element is not a symbol by another name, but rather a fundamentally separate and distinct ontological type.

But still, why not subsume two distinct ontological types under the single concept called "symbol"? The nature of machine thought it not clear. Much of the conceptual territory is obscured. In such a situation of incomplete understanding, conflating different basic ontological types together is usually considered unwise.

So the answer to the earlier question, Are computers purely symbol-manipulating devices? is: No. They can (and often do) create, modify, delete internal connectivity elements which are fundamentally different in nature and utility from symbols. And there is no good reason to conflate symbols and connectors together. And there is a good reason not to conflate them together.

Effect of the room's ontological deficiency

What does this ontological deficiency of the Chinese Room, its lack of connective elements, mean for the CRA? Firstly, in a form in which the argument is often expressed, the CRA is unsound. The premise, c*omputers are purely symbol-manipulating devices*, is false. The concept of the computer as a symbol-processing machine is defective. In fact computers can also create and process relational elements, and often do. They can create them, apply them, change them, and also delete them.

Are computers purely syntactic devices?

The second version of the CRA premise under discussion is: *computers are purely syntactic devices*. We have now happily discovered an ontological type other than the symbol that computers can easily create and manipulate. It's accepted (at present) that symbols are purely syntactic. But are the relational connectors also purely syntactic? If so (and assuming the combination of two syntactic types is also purely syntactic) the second version of the premise might be true even though the first is false.

Are connectors partly semantic?

This broader version of the CRA premise of interest says, *computers are purely syntactic devices*. Are relational connectors purely syntactic? Well, their essence is to connect and thereby provide a conduit from data at one end to data at the other end. That is, their essence is their related two ends. But nevertheless why not call connectors syntactic? We need a way and a reason to decide.

The CRA sets syntax in opposition to semantics. Syntax in the room is shape. Semantics is meaning (of the shapes). These meanings are outside the room and inside the minds of the Chinese speaking human observers. (The man in the room speaks no Chinese so the meanings of the Chinese symbols are not inside him.) Whether meanings are taken to be knowledge inside an observer or the things symbol shape refers to in the external world (trees, cars, whatever), the meanings are not in the room.

(Searle takes meanings to be interpretations in the minds of observers, which location he indicates by: "*symbols, by definition, have no meaning (or interpretation, or semantics)... except insofar as someone outside the system gives it to them*"[25].)

If connectors are purely syntactic then they will have no semantic aspects, no intrinsic semantic "content", they will not in themselves mean, refer to, or denote anything. They will not intrinsically, in and of themselves, indicate anything.

On the other hand, if an inner item does have semantic content, it will indicate some aspect or characteristic of the external world. For instance, if inner items created as a result of sensing external objects were to be shaped like the external objects, say shaped like an elephant when looking at an elephant, then the inner items would have semantic content in that they would share a value of a certain property (shape) with the macroscopic external objects sensed. (Ignoring what sameness of value amounts to, here.)

Inherent semantics of inner connectors

I want to argue that connectors do indicate something in the external world. However, this indication is not an indication in the linguistic sense of having a linguistic meaning. That is, it is not via the intercession of a human mind that observes the shape and understands what it means, and it is not by sharing a value of a property with an external object.

Connectors don't refer to or denote elements of the environment. These are linguistic concepts. But they do say something about the external world. And when combined with symbols, the combination says even more. And when symbols and connectors from different sensors are related together according to a certain principle of association, the resulting structure constitutes knowledge. That's the idea that I want to suggest.

As to the claimed relation between inner connectors and the external world, connectors are related to the world in a different way from linguistic symbols, a way that does not involve interpretation. They are inherently related. They don't need something else in order to indicate the world. They don't need a meaning. The indication is contained within the connector itself and its causal history.

Sensors

Shortly, I'd like to discuss sensors in some detail. A key matter in relation to sensors is the streams of data units, which we are calling symbols for convenience, that they emit into the inner world. They do this as a result of environmental particles or waves of particles repeatedly impacting the sensory surface. (Forces such as gravity are ignored at present.)

A sensor can be conceived of as having two sides, one facing the environment, the other directed at the inner world. The sensor is a causal interface between the inner and outer realms. It transduces external particulate impacts into internal symbolic transmissions. This will be discussed shortly.

Sameness of the value of a property

For convenience, we say computers process symbols and react to their shapes. But the same issues that go for symbols go for the values of any property of any data unit that the machine reacts to. The central system has a means to regard symbols of very similar shapes as being of the same shape. In fact each particular symbol will almost certainly be of a slightly different shape. But the system allows a slight leeway and ignores slight differences. The same goes for sensors when they create symbols and emit them into the inner world. Each symbol will be only approximately the same. The central system and the sensor need to allow adequately similar leeways, otherwise what a sensor regards as same shapes might be regarded by the central system as different shapes, or vice verse, and this could easily cause chaos.

Data in transit inside computers usually takes the form of clocked voltage levels, and in this regard there are two logic states named "0" and "1". The machine might treat all

voltages between say 0.3 and 0.8 volts as being one of these two levels, and between 3.2 and 3.7 volts as being the other level. Even though the actual levels in the wires fluctuate slightly, the device by its design treats them as being either one level, or logic state, or the other. In this way the machine through its design establishes sameness when in fact almost certainly no two units of substance processed by the machine are exactly the same.

Sensors

If a machine is going to understand its environment, then it will do so by way of the operation of its sensors.

Sensors, we may say, detect values of properties of impacting environmental particles or waves of particles. At least this is so for the sensors we are initially interested in. In response, the digital sensor emits symbols into the inner realm.

A way of thinking about this is that the sensory surface is divided into a number of detectors each of which reacts to a relatively small range of the values of the property that the sensor as a whole reacts to. Impacting particles whose values fall within a such given small range causes the respective detector, via the designed causation of the sensor, to emit into the inner world a symbol of a given shape. A reset period applies, and if the condition at the sensory surface is still within the small range once the reset period has expired, the detector reacts again and creates another symbol of the same shape and emits it into the inner world.

For example, if an olfactory sensor detects molecular shape, then a given receptor will react the same way to molecules of approximately the same shape. To a given receptor, sufficiently similar shapes will be treated as the same shape and yield the reaction. In the biological case, a receptor might have evolved to detect one type of molecule, but other types sufficiently similar in shape will trigger the receptor, leading to cheap knock offs of expensive perfumes, and components of artificial flavors.

Analogue sensor elements, such as bi-metal strips and noise-sensitive crystals, emit analogue signals and typically these are quantized into discrete data units by a sampling process. The unique quantization values can be regarded as the unique detectors.

The traditional human senses of vision, sound, taste, and touch are just a small subset of all human senses which include proprioceptive senses (mainly related to muscles condition and limb position) and various other interoceptive (internal bodily state) senses. The traditional senses are exteroceptors, receive sensory stimuli from outside the body. Interoceptors receive internal sensory stimuli related to blood pressure, oxygen level, states of the digestive tract, and more. Effector action isn't considered at present.

Togetherness in time

When one external condition at the sensory surface is followed by another there is a certain temporal relationship between the two conditions. One follows the other in time. When two symbols are emitted into the inner world one after the other in response to the detection of the two external conditions, the symbols are related by an instance of the same relationship. One follows the other in time.

The terms of the two relationship instances are different. The external terms are typically impacting external particles or waves of impacting particles such as air molecules or photons (the sensor detecting such properties as frequency of vibration, amplitude of

vibration, or molecule shape). The inner terms are typically clocked voltages, groups of electrons traveling down a wire or photons down a glass thread, which for convenience we are calling symbols. But the relationship *per se* between inner and outer is the same in both cases – adjacency in time.

Further, the outer instance causes, via the sensor, the inner instance. The reaction of the sensor to external togetherness in time of conditions at the sensory surface is to create an internal instance of the same relation – adjacency and sequence in time, temporal contiguity – but now between symbols.

The existence of the inner instance indicates the just-prior existence of the outer instance at the sensory surface. This is because the only way an inner instance can come to exist is by an external instance just previously existing and being detected by the sensor.

Recording temporal contiguity

The sensor sends a respective stream of symbols to the central system, and these symbols then arrive one after the other. Their adjacency in arrival time can be recorded by the system using connectors. These store the fact of temporal adjacency between the symbols as they arrive.

It records, for example, the fact that A followed B (and not C, D, E,...). The temporal relationship itself can't be stored. Any attempt to store it destroys it. The temporal relation embodies the absence of permanence. A connector confers permanence and is structural. So a connector can be stored in the place of adjacency in time as the symbols enter the system.

Such incoming symbols could simply be stored next to each other. If A follows B into the system, A could simply be stored next to B. In this case, spacial adjacency records temporal adjacency.

On the other hand, the fact that A followed B into the system could be recorded by connecting the two symbols together. This is another way to records their arrival one after the other in time. The connector could also record which arrived first. Such a connection could be created then applied no matter where A and B are stored (given the connector is long enough). For various reasons, the method of using connectors is far preferable to simply replacing temporal contiguity with spatial contiguity.

Example

Suppose a tone of A# is detected by a sound sensor, and in response the sensor sends a symbol of shape B to the central system. Then in the environment at the sensory surface, the A# is followed by an Eb. The sensor detects the Eb and dispatches a symbol of shape A to the central system. There is adjacency in time between the external conditions A# then Eb and between the emitted symbols B then A. Temporal contiguity between waves of impacting substance, air particles, at the sensory surface is mirrored as temporal contiguity between respective inner substance – symbols. The sensor duplicates the outer relation, but the terms, what is related, are different in both particularity and type.

The point being that the fact of the inner instance indicates the fact of the outer instance. However, the shape of the inner symbol is defined by the design of the sensor, and does not identify, refer to or denote the external condition that caused it.

Reaction of the central system to sensory streams

A central system can react to the shapes of arriving symbols. It can also and independently react to the fact of their adjacency in arrival time. The machine can create a physical connector and connect two temporally contiguous symbols together, without reacting to their shapes.

If syntax is shape, and if syntactic processing is reaction to shape, then in creating such connections, the machine doesn't respond to the syntax of the arriving items, and hence doesn't process them syntactically. But it does process them. It receives them, it stores them, and it connects them together.

The Chinese room can process input symbols associatively, as noted earlier. One symbol drops from the slot. The man follows the rules in the book. He puts this symbol in a basked on the left of the slot. The next symbol drops from the slot. He places it in a basket on the right of the slot. This alternation continues. Shape is irrelevant.

(If syntactic processing is manipulation contingent on shape, then even without having connectors, it seems the room is not a purely syntactic device.)

The semantics of temporal pairs

Two symbols arrive one after the other and are connected together. There are three items here: the two symbols and the connector. The connector is the permanent expression of temporal adjacency in the arriving stream – and earlier temporal contiguity in the environment at the sensory surface.

The relationship itself is a relation of time, however the moving instance of the relation and the stored record of the moving instance still contains shapes: the shapes of the symbols related. The shapes are still there. The *content* of the record is the two shapes and the connector. But in making this recording, the machine does not need to react to the shapes.

Having inner semantic content means intrinsically indicating something external. But do the inner records of contiguous arrival time indicate anything? The inner connector doesn't represent or denote some substantive object in the environment (as symbols via linguistic interpretation of shape do). And it was caused by, via a sensor, an instance of the same relationship at the sensory surface.

But the inner symbol pair doesn't indicate the terms of the external instance. For instance, they don't indicate that the external conditions were impacting waves of air molecules or that the sensor reacted frequency and amplitude of the waves. And the pair doesn't indicate even that there was an external instance. The relation between the inner and outer pairs is causal, and the causation goes from outer to inner.

Yet when instances of the same two symbol shapes arrive again at the central system one after the other from the same sensor, the fact is that the same two external conditions pertained again, one after the other, at the sensory surface. I think it can be said that in this case, the inner pair do indicate something, though perhaps not much – but still *something* – about the environment.

Yet perhaps an obvious rejoinder is, sure, this is *something* about the external world, but it's hardly enough to constitute knowledge. In fact it's woefully inadequate.

But is it? Indicating sameness of condition at the sensory surface might not seem enough, but could inner shapes and their contiguity in time be enough? Could they be sufficient components to build an inner semantic structure?

Types of causation

Two broad types of causation are noted: designed and natural.

Designed causation includes the causation of the sensor by virtue of which external particles react with sensory detectors in the sensory surface, and by virtue of which the sensor then emits respective symbols into the inner world. Designed causation includes the causation of the central system that receives these symbols then reacts to their proximity in time as they arrive and records the symbol pairs.

An example of natural causation is a rotating wooden block and a light source. Together they cause the photons that impact the sensory surface, and the sequence of reflections from the rotating block contain the changing values of the properties of the photons.

Another example of natural causation is that once inner representations of types of external macroscopic objects have been created in the inner semantic structure, repeated activation of two of these representations one after the other, indicates an external causal relation between objects of the represented types, for example between fire and smoke.

Implications for the CRA

The second version of the CRA premise under consideration says, *computers are purely syntactic devices*. We are now in a position to make a judgment about the truth of this premise. It appears to be false. The ability of computers to react to a relationship, instances of which say something (even if very little) about the external world, indicates that computers are not purely syntactic devices. If this is right, then Searle is wrong when he says "*...a digital computer is a syntactic machine. It manipulates symbols and does nothing else*"[26]. And since this claim is central to the CRA, the CRA is wrong, too.

Implications for AI

However, the inner symbol pairs of interest don't say what were the sub-microscopic external particulate conditions at the sensory surface, let alone what were the macroscopic objects further away. But even so, is there enough here, symbol shape and record of togetherness in time, to build knowledge? This inventory might insufficient, but is it?

The CRA concludes that it's theoretically impossible for a computer to have an inner semantics. Some of Searle's concepts are faulty, as noted. But he merely uses AI's concepts. He uses AI's concepts against AI. But he doesn't actually challenge the concepts. Rather, he happily accepts them. He accepts that computers are purely symbol-manipulating devices. He accepts that computers are purely syntactic devices. And so these defective ideas find expression as premises of the CRA and make the CRA unsound.

Bur having *some* semantic endowment, to use Minsky's term, is one thing. Exactly how could a computer possibly develop fully fledged human-like knowledge? The door to an intrinsic semantics in a computer, so called, might now be open a crack, but it seems that much more needs to be discovered about how to make a computer think.

Will computers think?

We assume that internal representations, in some form, can exist inside a computer of external objects and object types. These inner representations having been caused by respective external objects. And further, that these representations give a feral system an understanding in some sense of its environment, being a sense related to survival.

Consciousness might be another issue. Is consciousness needed for survival? Whether it is or not, it's typically regarded as part and parcel of thinking. I can suggest ideas about algorithms of consciousness, but this is a later issue. The key question at present is how could a computer develop inner representations from the components of substance shape (values of a property) and relational connectors, using simple operations such as matching, counting matches (i.e., repetition), and application of a count threshold?

Not enough semantics for knowledge?

In other words, how could the machine factor up repeated instances of inner shapes received next to each other in time into "representations" of external objects and object types (and other elements of knowledge, e.g., memory of actions, understanding of verbs, and the concept of ennui)?

This is a very major topic. But I think it's possible for computers to do this. Of course, if all the machine gets is sensory streams, and the machine will think, and it contains no knowledge at the start, then the sensory streams (baring ESP[27]) must contain all that's needed for an algorithm to build internal knowledge. I've done work on possible algorithms and structures. This is a large technical topic. It seems appropriate to state a principle, then consider how it might be applied.

Principle

The idea is that repeated symbol pairs, while seemingly insufficient for knowledge, can in fact be factored up to fully-blown semantic structures that allow a system to understand its world. The principle being:

> *All knowledge gained through experience is reducible to repeated instances of the relation of temporal contiguity.*

I believe that application of this principle can lead to human-like semantic content in a computer. The reason why I strongly believe this is a bit hard to explain, and anyway is probably completely irrelevant.

Generality

Most importantly, the above principle addresses the problem of generality. The most often repeated is the most general. The first structures built are the most general. With further application of the principle, progressive structure is added. The most general is no longer added because it is already there. What is added is the next most general, then that is there, and so the process continues from most general to most detailed – the *opposite* direction to a human programming a computer with a computation of the environment[28]. This is what's important. Computation starts with the most detailed. Association starts with the most general. That's why association can solve he problem of generalization – and why computation can't.

Now most learning has taken place. As sensory input arrives, the most general structure is activated first (because it was created first), then progressively more sensory detail is activated with further sensory input to the machine. If there is no more detail (from a given object or situation) in the sensory input or in the structure, the general understanding is still there. No novel situation is totally novel.

The lack of this characteristic, lack of most-general-to-least-general processing, in today's self-driving vehicle AI systems is what makes the systems so dangerous, and shows why there has been such failure to deal with "edge cases".

Possible structures

Suggestions can be made about the needed inner semantic structure. Almost certainly it will be a forest of data trees. There will be property trees (meaning trees caused by values of a property of impacting particles at the sensory surface), for instance caused by the color yellow. There will be object trees comprising a number of property trees as sub-trees. There will be a separate forest area per sensor, and a given such area will contain the property trees derived through the respective sensor. There will be only one instance of each symbol shape in a sensor area. There will be only one instance of a tree caused by a given value of a given property of impacting particles. A given property tree might be common to many object trees. Trees may not be binary. Binary means nodes have two or no children. Data tree nodes can be "fired" by a statistical trigger. Much more can be said about all this and appropriate algorithms.

Patterns of difference

Possibly a seeming problem is how to account for the richness and diversity of human knowledge. A structure of shapes, nodes and connectors seems extremely insufficient. Yet on the other hand, a structure of cellular tubes, neurons and unary pulses initially from biological sense organs seem highly improbable. If unary electric pulses moving inside different organic filaments is adequate for the organic brain, perhaps groups of different voltage levels moving serially through one wire (or a few in parallel) might be adequate for the machine brain. If structures embodying such difference are adequate for a semantics, then knowledge of external object types and relations between them could be regarded as patterns of difference. The idea of patterns of difference comes in part from elements of G. Spencer Brown's 1972, *Laws of Form*. So what else is there to knowledge other than patterns of difference derived from the environment via multiple sensor causality? I believe the answer is, Nothing.

Future of AI

It seems hard or impossible to imagine how a simple principle of repeated temporal association could account for knowledge. But if knowledge will come from a computer's sensory detection of its environment, or from a human's sensory detection of their surroundings, it will come from the properties of the units of substance that sensors emit and that multiple sensory streams contain and from relationships between them.

AI's present concepts are heavily influenced by the historical human use of the machine AI seeks to imbue with intrinsic intelligence. Some of these concepts are inadequate for the task, as indicated. In this sense, Dreyfus seems right. AI shares some characteristics with alchemy[29]. The conceptual discoveries of chemistry came from the theory, equipment, and empirical procedures of metallurgical alchemy in the political context of scientific freedom, plus the lust of princes for gold to finance military adventures. Most of this

financing of alchemical research was wasted, but the groundwork for chemistry was laid, and chemistry wouldn't have happened without it.

Equipment developed during WWII, the early computers, established the technical basis of the research field of AI. Since then, most AI funding has been military with the goal of reducing the cost of military adventures, both political and financial. Most of this taxpayer funding, gifted mainly by the US government without taxpayer consent, was wasted (in terms of the original research goal, that is – a computer with a human-like general intelligence). It's yet to be seen whether AI theory and practice along with computer equipment have set the foundations of human-like general intelligence in a machine.

But we ought to proceed on the assumption that they have. My argument is that, as with alchemy, some AI concepts are inadequate. And that's the real message of the CRA.

Space and time

The idea is that instances of the relation of temporal contiguity in symbol streams might contain the building blocks of knowledge. Symbols are made of substance. Substance has values of one or more properties. Some systems can react differently to such value differences in what they process. Substance is extended in 3-dimensional space.

One would think that for a system to intrinsically learn about the world, it would need to fundamentally react not only to the content of extended space, but also to temporal relations. How else, for example, might it understand the meanings of verbs?

The present text indicates that there is a sense in which computers can react to both space and time. By reacting to the relation of temporal contiguity and then building structure that contains the shapes of the related items, computers may have a fundamental ability, given adequate algorithms, to learn about the world.

Conclusion

In his Chinese Room Argument, John Searle presents a picture of the computer as fundamentally incapable of learning. This comes from the computational idea that machine behavior (more generally, causation) is defined by human design, and a change in behavior requires a change in design. The program of the stored-program computer provides a very convenient and quick way for human modification of the causation of the machine. But any "learning" then requires human specification of the new knowledge or behavior. Turing also struggled with this problem. In the CRA case, the machine has no way to intrinsically learn anything, including Chinese.

Searle uses the concept of computation to understand computer operation as execution of human-prescribed rules contingent on shape, which rules do not reference the meanings of the shapes. Searle calls this type of processing formal, or syntactic. The Chinese room thought experiments presents this conception of the computer.

In this conception, the ontology comprises only units of substance, called symbols, and values of a property of the substance, regarded as shape. Hence the computer processes what it processes by reacting according to human-created rules about the shapes of the things that it processes.

The ontology of the Chinese room is deficient. One would expect a device that could be a human-like mind to fundamentally react to elements of both space and time. The Chinese

room reacts to elements only of space – substance called symbols which are extended in 3-dimensional space. However a computer (so called) can also react to time. Specifically, to the temporal relation of adjacency in time between moving symbols as they enter the system. This fact of the temporal relation between arriving units of shaped substance can be recorded in storage as permanent connective elements. Being relational, such connectors are structural.

Analysis shows that connectors are not symbols by another name but rather a distinct ontological type. These connectors can build inner structure by connecting the substance of symbols or places where symbols are stored. An arriving sensory stream thus provides not one ontological type, symbols, but two ontological types, symbols and connectors, which can then be components of populated structure. Such a structure derived from incoming sensory streams can indicate aspects of the external world and hence has some semantic content.

On the face of it, such semantic content, while sufficient for rebuttal of Searle's Chinese Room Argument, seems very insufficient for inner representation of types of external macroscopic object. However, further examination reveals a number of computer operations including matching, counting matches and applying a threshold match count, that offers an indication that computers may be able to factor up this seemingly insufficient semantics to a comprehensive representational semantics, enough for knowledge of external macroscopic objects and relations between them. If computers will one day think, baring Turing's ESP, sensory streams will be the fountain of all knowledge gained through experience, and the units of substance the streams contain plus relations between them is the only content of the fountain.

Generality is seen as fundamentally anti-computational in that in computation the human writes a program which must then account for all possible situations the machine might encounter, which for games such as checkers and chess is tractable to some extent, but for the environmental situation is extremely intractable and leads quickly to the frame and other severe problems. A connective structure is fundamentally relational. One built on a principle of repetition first stores as populated structure the most often repeated input elements, and the most often repeated is the most general, hence such a connection structure begins with the most general and with learning achieves detail, whereas computation begins with the most detailed, and generality is never achieved except to the extent of accumulated detail plus search, or heuristic guesswork.

In short, the idea of using connective elements derived from sensory streams to create structure populated with the substance of sensory streams on the basis of matching and repetition count is a non-teleological idea of inherent learning where the program does not dictate inner structure on the basis of human interpretation of input shape. Computation is human-created rules about shapes which have a human interpretation irrelevant to the formality of the computing process. Whereas the associative process of building structure is a fundamentally different (and much faster) ball game entirely, and the pattern of shapes in the structure depends on the sensed environment.

If this basic associative structural approach to artificial general intelligence is useful, a key question is no longer, How could a computer overcome the CRA's conclusion of the impossibility of inner semantic content and knowledge? Rather, the question now is, How could the two components of sensory streams in concert be factored up by one or more human-created simple algorithms to build semantic structures that represent types of external macroscopic objects and relations between them? In this regard, the concept of patterns of difference is suggested.

Endnotes

1 P. J. Hayes and J. McCarthy, 1969, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", in Bernard Meltzer and Donald Michie (Eds), 1969, *Machine Intelligence 4*, 463-502.

2 Quoted by Harnad in Stevan Harnad, 2001, "Minds, Machines, and Searle 2: What's Right and Wrong about the Chinese Room Argument", in Mark Bishop and John Preston (Eds), 2002, *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*, Clarendon Press, 294-307.

3 David Cole, 2004, "Chinese Room Argument" *Stanford Encyclopedia of Philosophy*, available online at plato.stanford.edu.

4 John R. Searle, 2014, "What Your Computer Can't Know", *New York Review of Books*, October 9, 2014, available online.

5 John R. Searle, 1980, "Minds, Brains and Programs", *The Behavioral and Brain Sciences*, vol. 3, 1980, Cambridge University Press, available online.

6 John R. Searle, 2002, "Artificial Intelligence and the Chinese Room: An Exchange [with Elhanan Motzkin]", *New York Review of Books*, vol. 36, page 45. The respective passage is quoted in John Preston, "Introduction", in John Preston and Mark Bishop (Eds), *Views into the Chines Room: New Essays on Searle and Artificial Intelligence*, Clarendon Press, Oxford, 2002, page 35.

7 John R. Searle, 1984, *Minds, Brains and Science*, Harvard University Press, page 34, available online.

8 According to Pylyshyn (Zenon Pylyshyn, 1984, *Computation and Cognition*, The MIT Press Paperback Edition, 1986, page 62): "To count as a computation [the computer] must contain symbols that are interpreted. In other words, the symbols must represent numbers, letters, or words, etc. (the slogan here is: "no computation without representation"). ... The [interpretation] is provided by a person who takes the symbol to be about something – that is, the person gives the symbols a semantic interpretation".

9 A. M. Turing, 1936, "On Computable Numbers, With an Application to the Entscheidungsproblem", available online, page 231: "We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions q_1, q_2, ..., q_i, which will be called 'm-configurations'. The machine is supplied with a 'tape' (the analogue of paper) running through it, and divided into sections (called 'squares') each capable of bearing a 'symbol'"; and on page 249: "Computing is normally done by [a human] writing certain symbols on paper"; and further, on page 250: "The behaviour of the computer [human performing a computation] at any moment is determined by the symbols which he is observing, and his 'state of mind' at that moment".

10 A. M. Turing, 1950, "Computing Machinery and Intelligence", *Mind*, vol. LIX, No. 236, 433-460, available online. To Turing the behavior of a computer is fully "described" (defined) by its programming. Hence, "How can the rules of operation of the machine change? They should describe completely how the machine will react whatever its [past, present or future] history might be" (page 458).

11 A. M. Turing, 1950, "Computing Machinery and Intelligence", *Mind*, vol. LIX, No. 236, 433-460, available online: "The idea of a learning machine may appear paradoxical to some readers. How can the rules of operation of the machine change? They should describe completely how the machine will react whatever its history might be, whatever changes it might undergo. The rules are thus quite time-invariant. This is quite true. The explanation of the paradox is that the rules which get changed in the learning process are of a rather less pretentious kind, claiming only an ephemeral validity".

12 A. M. Turing, 1951, "Intelligent Machinery, a Heretical Theory", typed transcript of radio discussion notes for BBC program, *The '51 Society*, reproduced in B. Jack Copeland (Ed.), 2004, *The Essential Turing*, Oxford University Press, 472-475.

13 A. M. Turing, 1950, "Computing Machinery and Intelligence", *Mind*, 49: 433-460, available online.

14 W. S. McCulloch and W. Pitts, 1943, "A Logical Calculus of Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics*, vol. 5, 115-133, 1943; and D. O. Hebb, 1949, *The Organisation of Behavior: A Neuropsychological Approach*, New York, John Wiley & Sons: "When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased".

15 P. J. Hayes and J. McCarthy, 1969, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", in Bernard Meltzer and Donald Michie (Eds), 1969, *Machine Intelligence 4*, 463-502.

16 John McCarthy, 1959, "Programs with common Sense", in Marvin Minsky (Ed.), 1968, *Semantic Information Processing*, The MIT Press, 403-417, available online. McCarthy recognizes the problem of common-sense knowledge and proposes a solution to it couched in the concepts typical at the time.

17 Marvin Minsky (Ed.), 1968, *Semantic Information Processing*, The MIT Press, page 26 passim. Minsky vastly underestimates the difficulty of the problem of common-sense knowledge, but draws attention to it.

18 James Lighthill, 1973, "Artificial Intelligence: A General Survey", *Artificial Intelligence, a paper symposium,* UK Science Research Council, 1973, Section 3.

19 Steven Harnad, 1990, "The Symbol Grounding Problem", *Physica* D 42, 335-346.

20 John R. Searle, 1980, Ibid.

21 Turing's so called prediction was actually an appeal for the term "intelligence" to be redefined, presumably to mean behavior. He said in his 1950 paper, "...I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted". Presumably what he meant was that he hoped that his test of machine intelligence, now called the Turing test, a purely behavioral test, would become the *de fact*o concept of machine intelligence. And certainly, many in AI today see it that way.

22 A. M. Turing, 1950, "Computing Machinery and Intelligence", *Mind*, vol. LIX, No. 236, 433-460, available online, page 458.

23 Pointers are addresses, unique identifiers usually regarded as integers, of memory locations. The Chinese room might be able to implement pointers and hence implement relational elements, which if so would be extremely tedious. But this seems a separate issue. The point is that creation of relational elements, structure, and the ability to react to relational elements is absent from the conception of the computer that Searle and the CRA use. They are absent from Searle's description of the room as the essence of the machine. But the ability to react to relations between substance as well as to values of properties of substance is surely an element of essence.

24 John R. Searle, 2014, "What Your Computer Can't Know", *New York Review of Books*, October 9, 2014, available online.

25 John R. Searle, "Artificial Intelligence and the Chinese Room: An Exchange [with Elhanan Motzkin]", *New York Review of Books*, vol. 36, page 45. The quoted passage is in John Preston, "Introduction", John Preston and Mark Bishop (Eds), *Views into the Chines Room: New Essays on Searle and Artificial Intelligence*, Clarendon Press, Oxford, 2002, page 35.

26 John R. Searle, 2014, "What Your Computer Can't Know", *New York Review of Books*, October 9, 2014, available online.

27 John R. Searle, 2014, "What Your Computer Can't Know", *New York Review of Books*, October 9, 2014. Available online.

28 An explanation of this programming can be found in Daniel C. Dennett, "Cognitive Wheels: The Frame Problem of AI" in Zenon W. Pylyshyn (Ed.), 1987, *The Robot's Dilemma: The Frame Problem in Artificial Intelligence*, or in other comprehensive explanations of the frame problem or problem of combinatorial explosion.

29 Hubert L. Dreyfus, 1965, "Alchemy and Artificial Intelligence", RAND Corporation, publication P-3244, page 84: "Alchemists were so successful in distilling quicksilver from what seemed to be dirt, that after several hundred years of fruitless effort to convert lead into gold they still refused to believe that on the chemical level one cannot transmute metals. To avoid the fate of the alchemists, it is time to ask where we stand [about AI]".